

Grafos Encontram Redes: Aprendendo Representações Vetoriais para Vértices utilizando Redes Neurais

**Seminário Integrado de
Software e Hardware (SEMISH)
XLII CSBC
3 / 8 / 2022**

**Daniel Ratton Figueiredo
PESC/COPPE/UFRJ**



Grafos e Redes

- **Grafos:** abstração para codificar relacionamento entre pares de objetos
 - ferramenta matemática
- **Redes:** grafos gerados de dados reais
 - redes sociais, de informação, tecnológicas, etc
- **Ciência de Redes x Teoria dos Grafos**
 - dois lados de uma mesma moeda

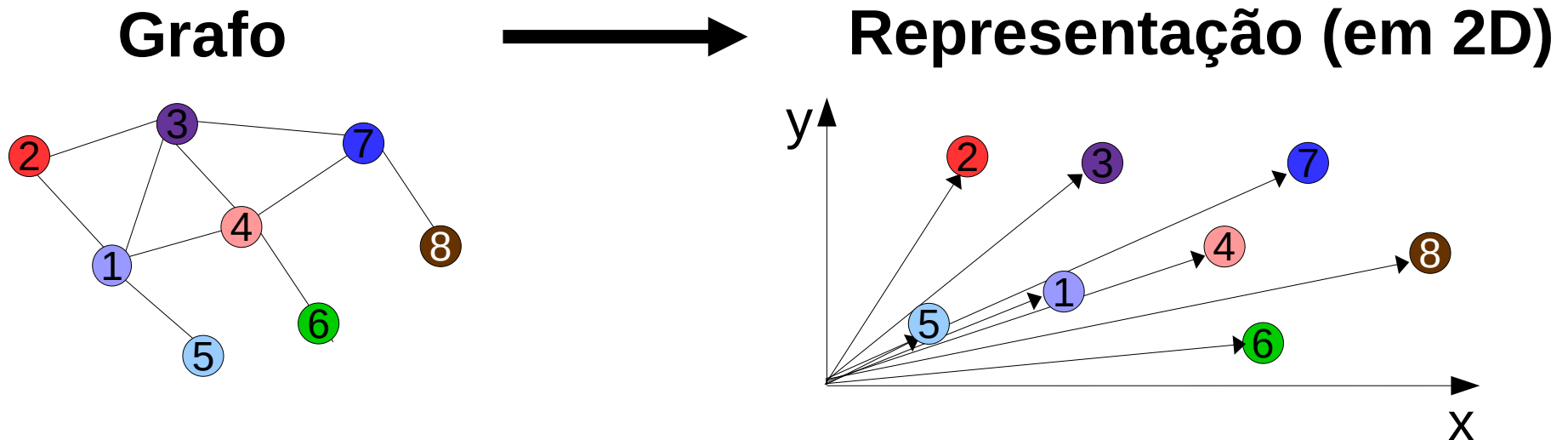
Fábio Botler

Grafos = Redes



Representação de Vértices

- **Problema:** mapear vértices de um grafo no espaço Euclidiano
 - cada vértice é associado a um vetor
 - vetor é a representação (*embedding*) do vértice



Por que fazer isso?
Como fazer isso?

Por que?

- Representação coloca o grafo no espaço Euclidiano
- Vértices se tornam vetores
 - permite uso de técnicas de espaço contínuo para analisar o grafo
 - ex: K-means para clusterização, K-NN para vizinhos mais próximos, similaridade de cosseno
- Vetores são atributos dos vértices
 - usado em algoritmos de aprendizado
 - ex: classificação dos vértices

Mais fácil trabalhar com vetores!

Ganhando Aderência



Stanford Computer Forum 2021 Affiliates Meeting

Stanford Graph Learning Workshop

10:45 - 12:00 Industry panel [\[Video\]](#) [\[Liv\]](#) 16:15 - 17:00 Industry panel [\[Video\]](#) [\[Livestream\]](#)

- Andrew Zhai, Pinterest
- Jaewon Yang, LinkedIn
- Benedek Rozemberczki, AstraZeneca
- Hatem Helal, Graphcore
- Nadia Fawaz, Pinterest (moderator)
- Kim Branson, GlaxoSmithKline
- Marinka Zitnik, Harvard University
- Naren Chittar, JP Morgan Chase
- Yu Liu, Facebook AI
- Hema Raghavan, LinkedIn (moderator)

IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, VOL. 2, NO. 2, APRIL 2021

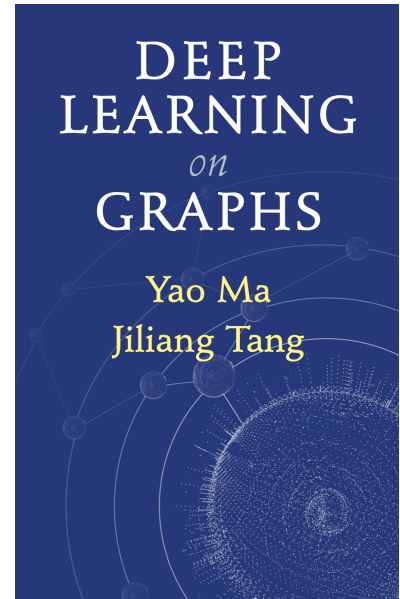
Graph Learning: A Survey

Feng Xia , Senior Member, IEEE, Ke Sun , Shuo Yu , Member, IEEE, Abdul Aziz ,
Liangtian Wan , Member, IEEE, Shirui Pan , and Huan Liu , Fellow, IEEE

184 referências !

Learning on Graphs Conference

LoG is a new annual research conference that covers areas broadly related to **machine learning on graphs** and **geometry**, with a special focus on review quality.



Cambridge Univ
Press 2021

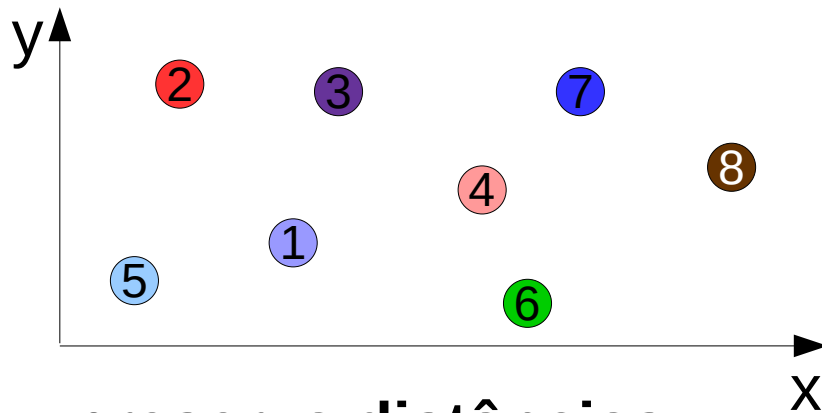
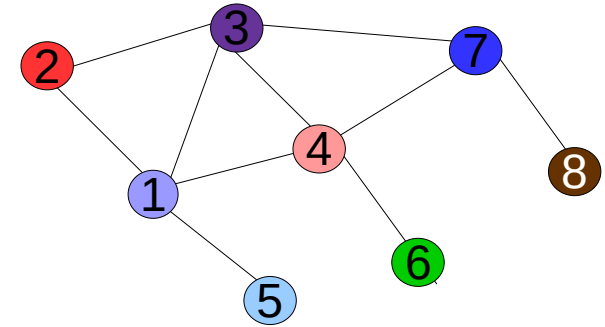


SEMISH 2022

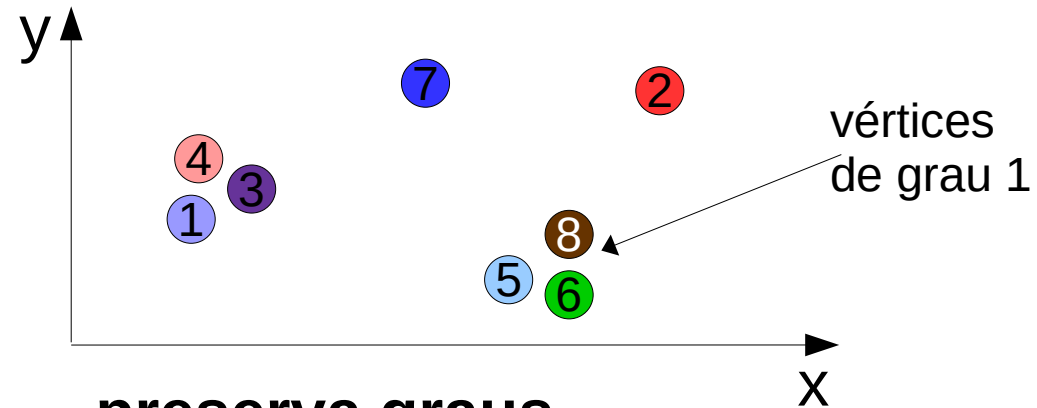
Boas Representações



- Qual a melhor representação para um grafo?



preserva distâncias
no grafo



preserva graus
no grafo

Depende dos objetivos!

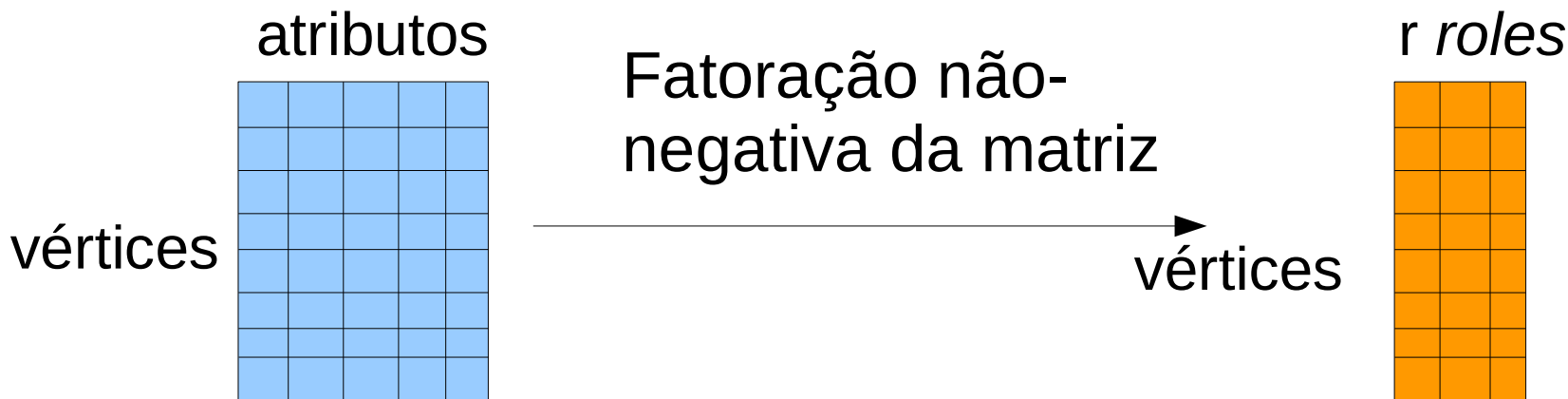
- Qual a finalidade da representação?
- Não há bala de prata!

Como ?

- Duas abordagens para gerar representações
 - 1) fatoração/otimização: mais antiga e menos flexível
 - 2) aprendizado: mais recente e mais flexível
- Muitas técnicas em cada abordagem
- Rede com atributos: vértices possuem atributos (não estruturais)
 - ex. nacionalidade, idade, gosto musical, etc
- Atributos usados para gerar representações
 - mistura estrutura da rede com atributos

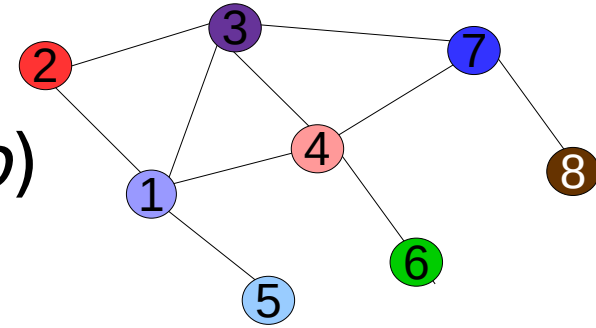
Rolx

- Matriz de atributos e vértices
 - estruturais (grau, triângulos, etc)
 - exógenos (nacionalidade, idade, etc)
- Centenas de atributos



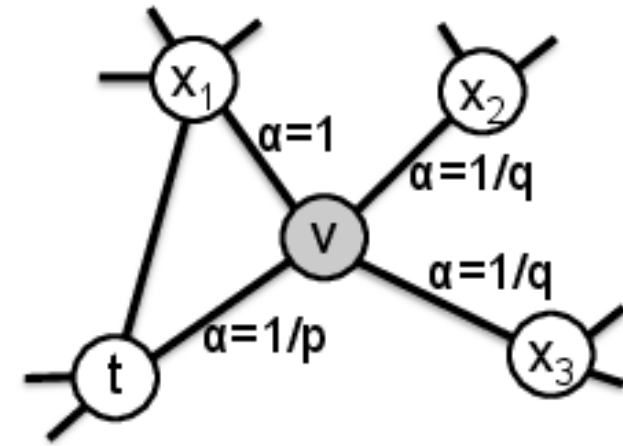
- Mapeia vértices em *r* papéis (*r* é parâmetro)
- Representação indica papéis do vértice

DeepWalk

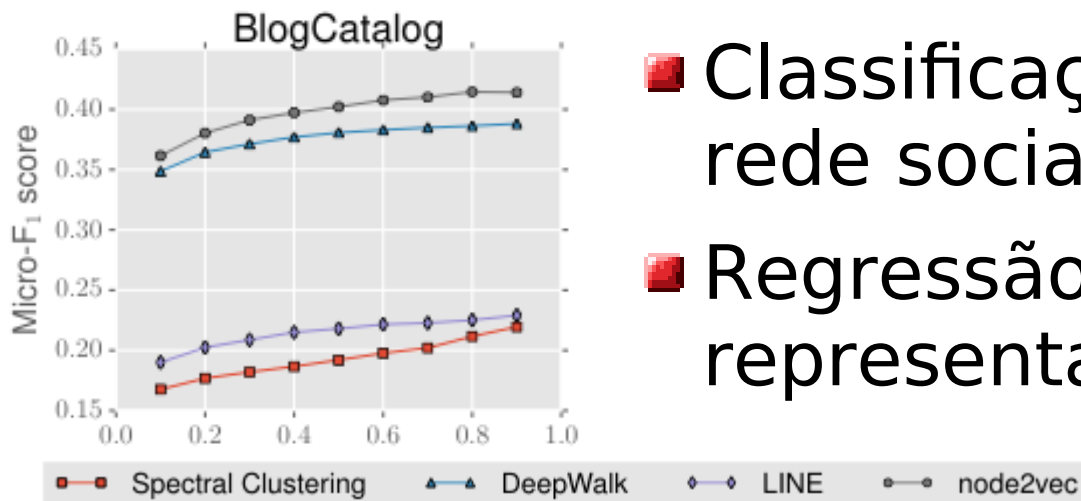


- Gera sequência de vértices (*contexto*) usando passeios aleatórios curtos
 - ex: 2,1,3,2,3 – 2,3,4,1,2 – 2,3,4,7,3, ...
- Contextos tratados como frases de uma linguagem natural
 - identificadores dos vértices são as palavras
- Usar *word2vec* para gerar representação para palavras (vértices)
 - maximizar probabilidade do contexto dado a palavra
- Vértices com contextos similares tem representações próximas

node2vec



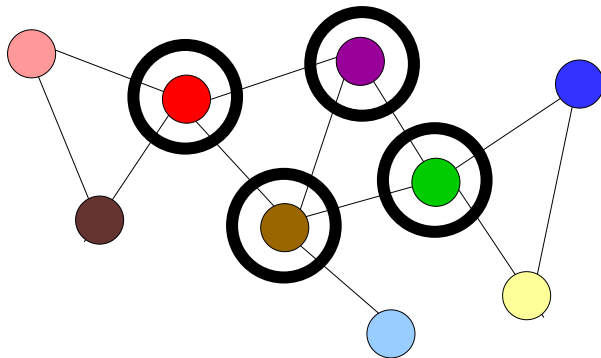
- Mesma abordagem que DeepWalk
- Contexto gerado por passeios aleatórios enviesados
 - permite voltar ou ir mais longe
 - parâmetros p , q controlam viés do passeio
- Enfatizar estrutura local quando necessário
 - ajustar representação para diferentes problemas



- Classificação de Bloggers usando rede social (10K nós, 39 classes)
- Regressão logística treinada com representações dos vértices

Identidade Estrutural

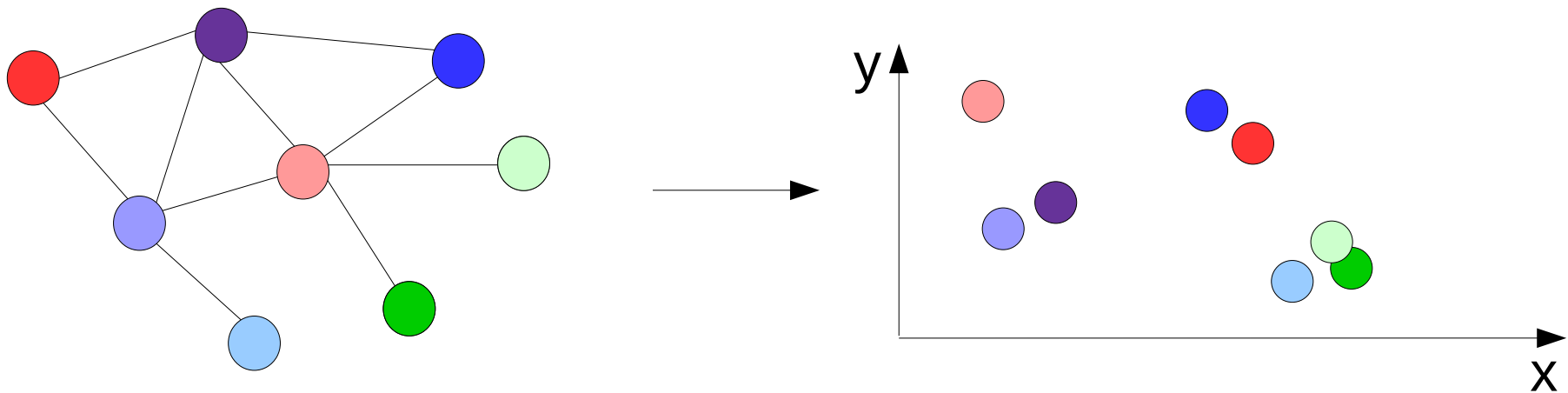
- Vértices da rede possuem diferentes papéis
 - pessoas, páginas web, proteínas, etc
- Identidade estrutural
 - identificação dos vértices baseado na estrutura da rede (e nenhum atributo)
 - relação direta com o papel do vértice
- Automorfismo: forte equivalência estrutural



- Vermelho, Verde: automorfismo
- Roxo, Marrom: estruturalmente parecidos

Ideia para Solução

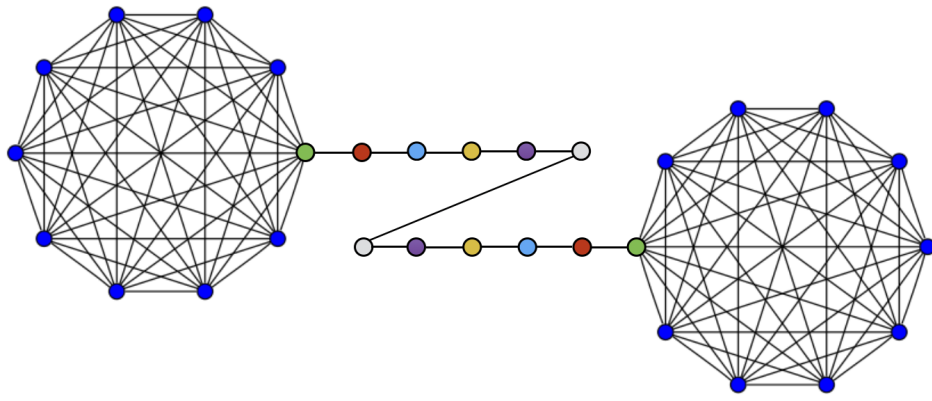
- Representar vértices no espaço euclidiano
- Vértices estruturalmente similares ficam próximos no espaço



- Usar técnicas de classificação/clusterização convencionais a partir da representação

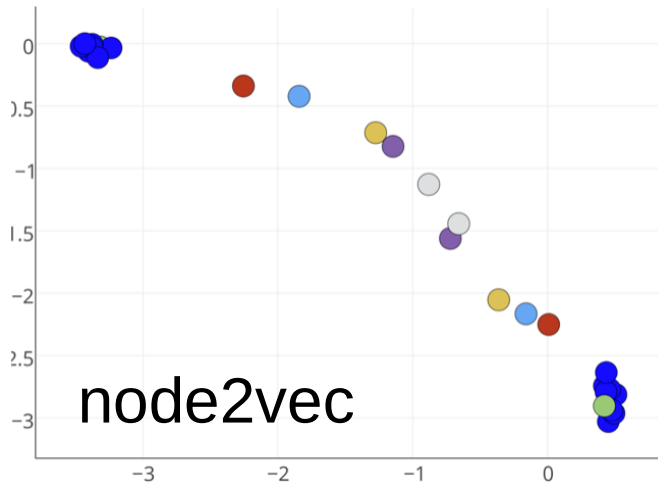
struc2vec

- Primeira técnica para atacar este problema utilizando aprendizado
- Construir um outro grafo: arestas indicam similaridade estrutural
 - não utiliza os rótulos dos vértices (apenas os graus)
- Gerar contexto usando passeios aleatórios enviesado no novo grafo (multi-camadas)
- Treinamento idêntico ao node2vec
 - geração do contexto é diferente
- Nova metodologia para avaliação

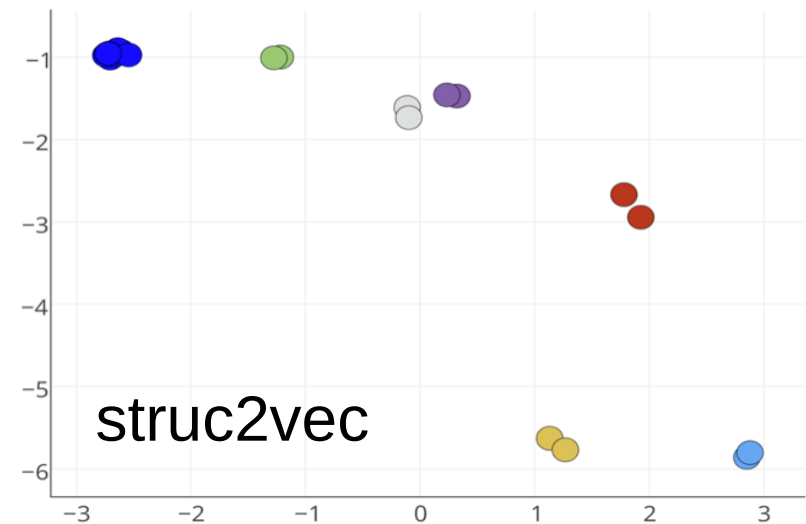


Exemplo

- Projeção em 2D das representações



- Captura proximidade no grafo

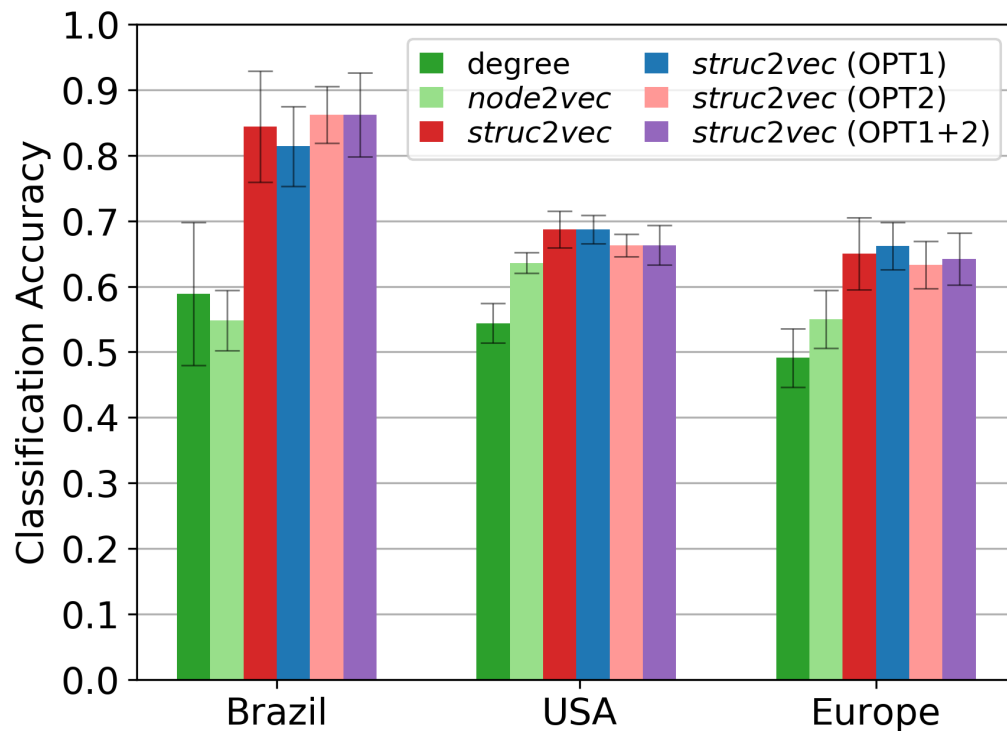


- Captura similaridade estrutural
- automorfismos bem próximos

Classificação de Aeroportos

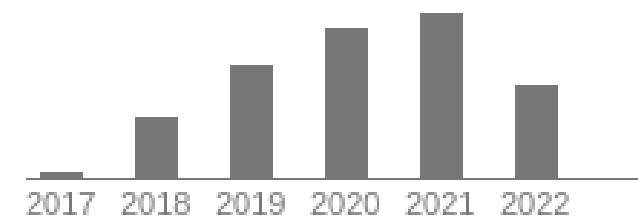
- Representação dos vértices usada para treinar um classificador (movimentação de aeroportos)

- 4 classes, regressão logística



- Artigo bem recebido pela comunidade

Total de citações Citado por 927



- 1o. lugar no CTDIAC

- tese de mestrado do Leonardo

Alerta de Troca de Tema

Graph Neural Networks

- Métodos anteriores
 - puramente estruturais, não utilizam atributos
 - baseados na geração de contexto (aleatório)
 - representação apenas para vértices da rede
- Vizinhaça e atributos são indicativos do papel do vértice
 - parecido com pixels de uma imagem

Graph Neural Networks

- Metodologia para gerar representações
 - inspirado em Convolutional Neural Networks (usado em imagens)

GNN: Principais Ideias

- Representação do vértice é função da representação dos vizinhos
 - ordem dos vizinhos não importa
- Iterativo: vértice possui uma representação para cada iteração/camada
 - representação inicial são atributos dos vértices
- Rede neural “transforma” a representação
 - representação de entrada em saída
- Função objetivo depende das representações
 - exemplos positivos e negativos

Formalizando

- $x_{v,k}$: representação do vértice v na iteração k
- $y_{v,k}$: representação (média) dos vizinhos de v na iteração k
- W_k, S_k : matrizes que transformam representação da iteração $k-1$
- σ : função de ativação
- N_v : vizinhos do vértice v
- x_0 : representação inicial (função dos atributos)

$$x_{v,k} = \sigma(h_{v,k})$$

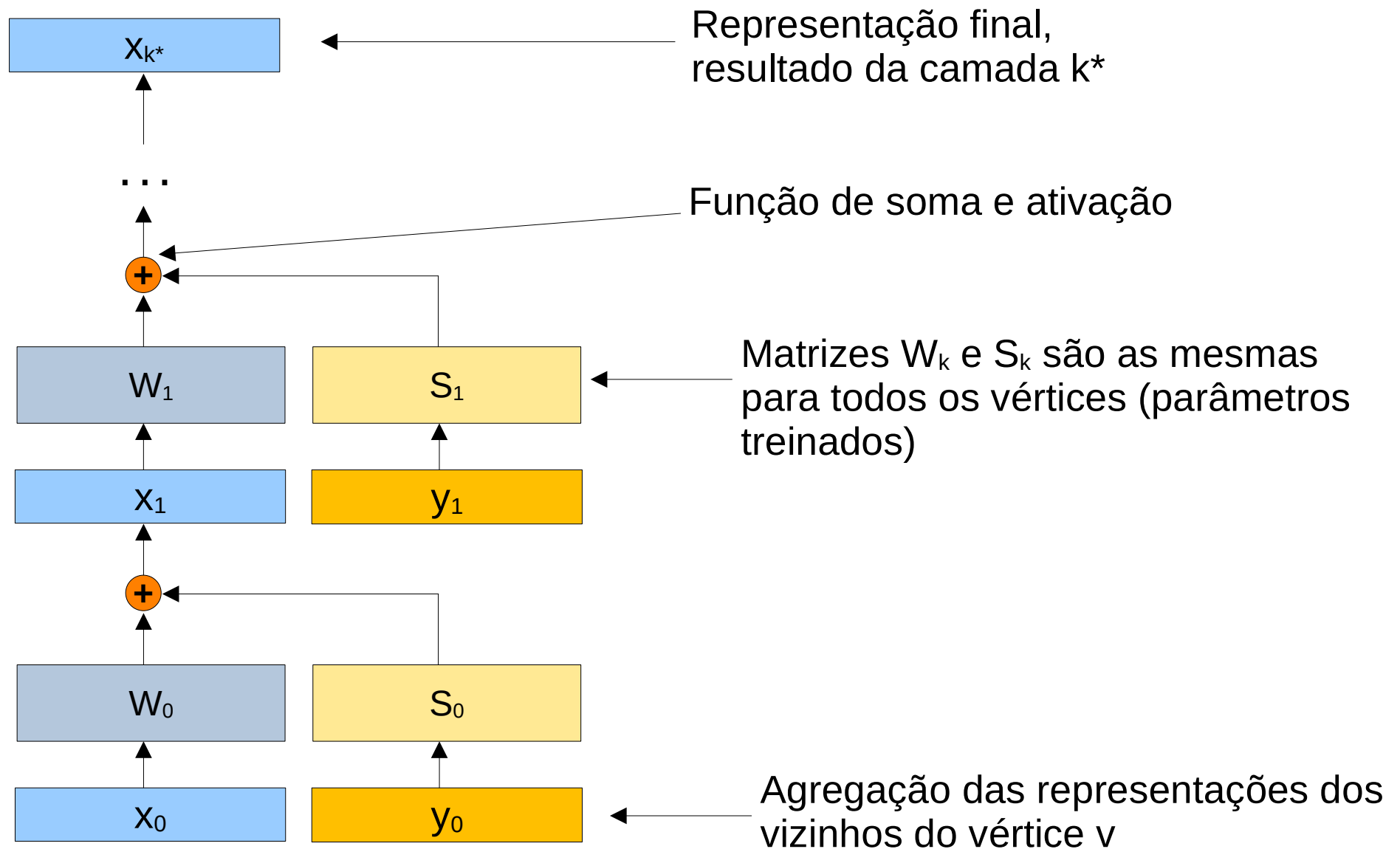
$$h_{v,k} = W_k * y_{v,k-1} + S_k * x_{v,k-1}$$

$$y_{v,k-1} = \frac{1}{|N_v|} \sum_{u \in N_v} x_{u,k-1}$$

Outras funções de soma podem ser usadas!

Outras funções de agregação podem ser usadas!

Diagrama da Rede Neural



■ Mesma rede usada para todos os vértices

Função Objetivo

- Em geral, *triplet loss*
 - amostras positivas, negativas, e margem
- Usando métrica entre representações
 - ex. distância Euclideana, produto interno

$$f = \sum_{(u,v) \in P} \|x_u - x_v\| - \sum_{(u,v) \in Q} \|x_u - x_v\| + \delta$$

Conjunto de pares positivos

Conjunto de pares negativos

Margem

- Diferentes formas de construir P e Q
 - P = pares vizinhos, Q = pares ao acaso

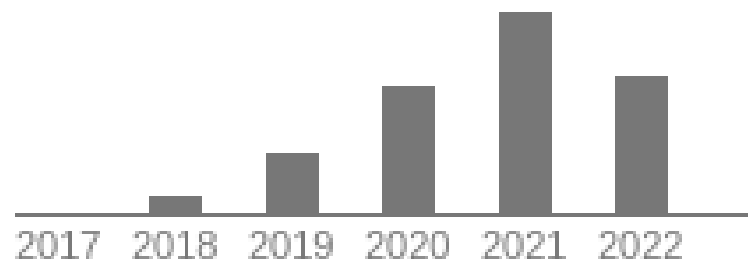
Características

- Não precisa ser treinada com todos vértices
 - amostragem, inclusive da vizinhança
- Rede treinada gera representação para qualquer vértice
 - mesmo vértices nunca antes vistos, ou vértices de outras redes
- Metodologia muito flexível
 - atributos de entrada, função de agregação, função de fusão, função objetivo, técnica de amostragem, número de camadas, dimensão da representação, etc

GraphSage

- Uma das primeiras GNNs
 - definiu essência da metodologia
- Abordagem rompeu com paradigma anterior

Total de citações Citado por 6968



Enorme sucesso!

- Centenas de variações e melhorias e usos

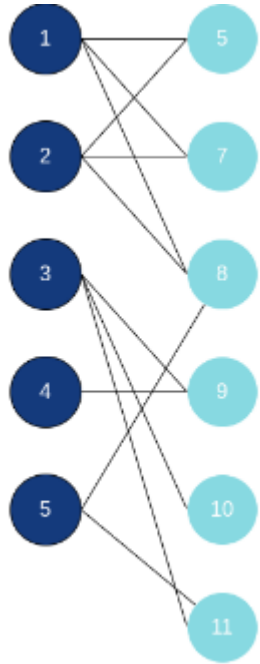
Classificação e Previsão

- Classificar vértices de uma rede
- Prever arestas de uma rede
 - 1) Treinar GNN para gerar representação dos vértices
 - 2) Treinar modelo de classificação/previsão usando representações

Muitas aplicações!

- Redes de proteínas, redes sociais, redes de documentos, rede financeiras, etc

Recomendação

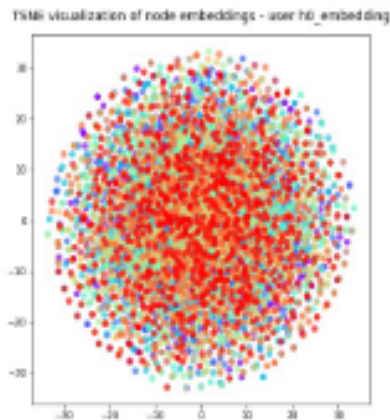


- Rede bipartida: usuários e produtos
- Classificação de usuários e produtos
 - baseado em atributos e estrutura da rede
- Classes usadas para fazer recomendação
- GNN para representação em redes bipartidas
 - espaço de usuários e espaço de produtos
- Vizinhança de dois saltos preserva tipo
 - metodologia com amostragem e pesos

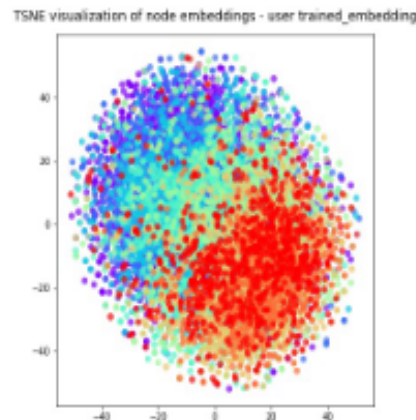
Exemplo

- Modelo SBM adaptado para redes bipartidas
- 8 classes de usuários, 8 classes de produtos
- Projeção em 2D (t-SNE) das representações
 - função do treinamento da GNN

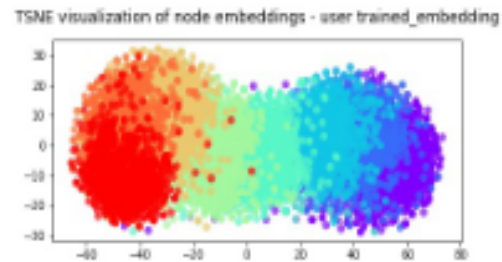
Inicial



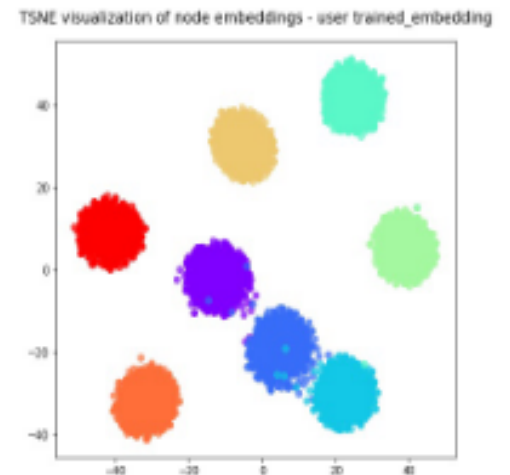
5 épocas



10 épocas



30 épocas

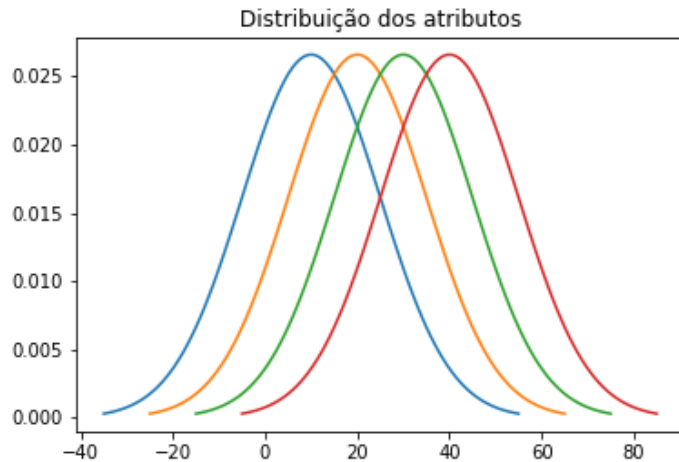


Clusterização

- Dividir a rede em comunidades (clusters)
 - função dos atributos e estrutura da rede
- Atributos são ruidosos (não determinam a comunidade)
- Metodologia iterativa (não supervisionada)
 - 1) GNN gera representações
 - 2) K-means gera clusters
- Clusters obtidos em (2) afetam “vizinhança”
- Representações obtidos em (1) afetam clusters
- Múltiplas rodadas de GNN!

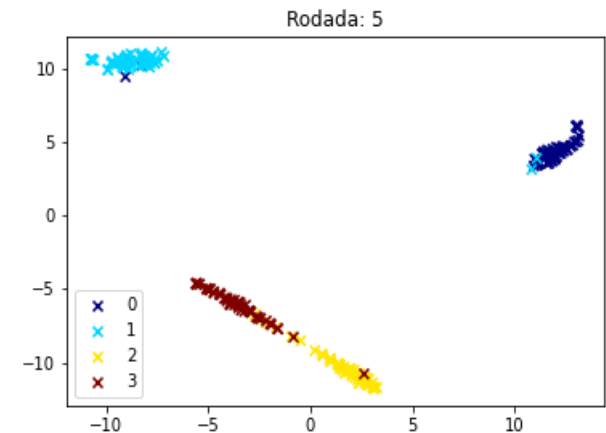
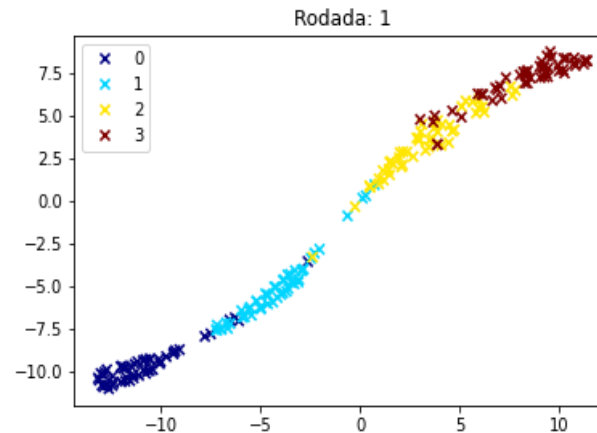
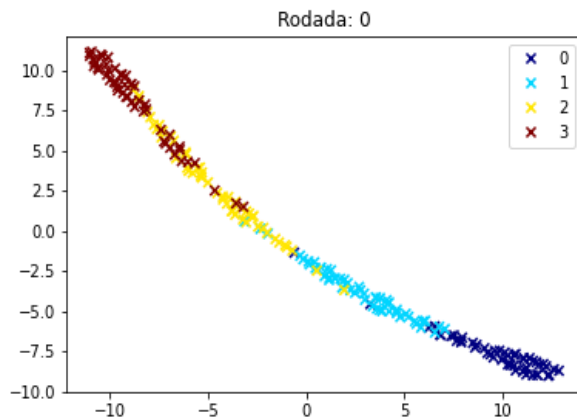
Exemplo

■ Modelo SBM com quatro classes, simétricas



■ Atributo: amostra de distribuição normal para cada classe

■ Projeção em 2D (t-SNE) das representações por rodadas



Algoritmos se ajudam!

Desafios

- Como gerar a melhor representação para seu problema?
 - artigos mostram que representação gerada por técnica A é melhor que B para problema X
- Como escolher atributos e codificar representação inicial?
 - artigos agregam atributos de forma arbitrária
- Como lidar com grafos que evoluem?
 - vértices novos, vizinhança evoluindo, etc
- Como lidar com *fairness* ?
 - diferentes classes igualmente representadas

Conclusão

- Graph Neural Network (GNN)
 - metodologia flexível para gerar representação vetorial para vértices misturando atributos e estrutura da rede
- Muitas variações, muitas aplicações
 - sendo testadas e usadas por diversas empresas

Ferramenta fundamental

- Do saco de ferramentas do cientista de dados

Obrigado!



- Dúvidas, perguntas, ou comentários?

- Contato

- daniel@cos.ufrj.br

- www.cos.ufrj.br/~daniel